MONGODB - DEPLOYMENT

http://www.tutorialspoint.com/mongodb/mongodb_deployment.htm

When you are preparing a MongoDB deployment, you should try to understand how your application is going to hold up in production. It’s a good idea to develop a consistent, repeatable approach to managing your deployment environment so that you can minimize any surprises once you’re in production.

The best approach incorporates prototyping your set up, conducting load testing, monitoring key metrics, and using that information to scale your set up. The key part of the approach is to proactively monitor your entire system - this will help you understand how your production system will hold up before deploying, and determine where you will need to add capacity. Having insight into potential spikes in your memory usage, for example, could help put out a write-lock fire before it starts.

To monitor your deployment MongoDB provides some commands that are shown below:

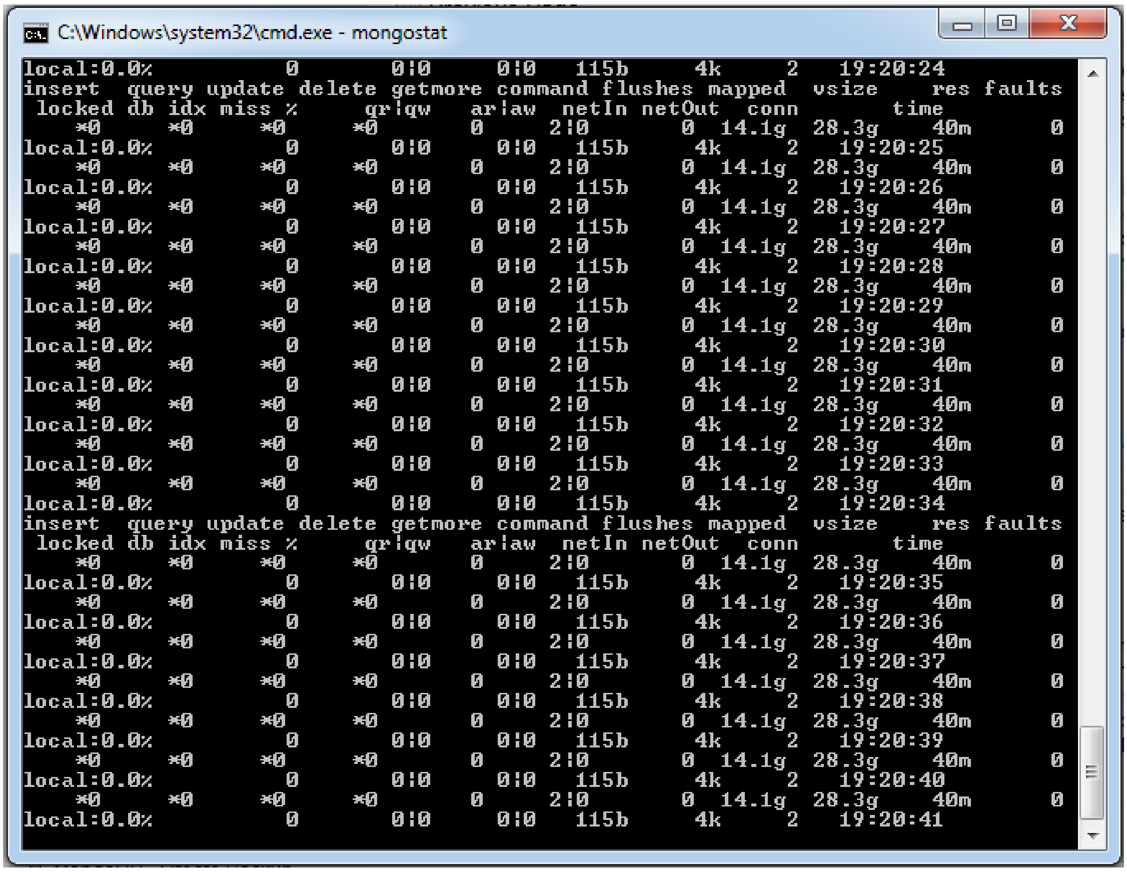
mongostat

This command checks the status of all running mongod instances and return counters of database operations. These counters include inserts, queries, updates, deletes, and cursors. Command also shows when you’re hitting page faults, and showcase your lock percentage. This means that you're running low on memory, hitting write capacity or have some performance issue.

To run the command start your mongod instance. In another command prompt go to **bin** directory of your mongodb installation and type **mongostat**.

```
D:\set up\mongodb\bin>mongostat
```

Output of the command is shown below:



```
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:24
insert  query update delete getmore command flushes mapped  vsize    res faults
 locked db idx miss %      qr|qw   ar|aw  netIn netOut  conn       time
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:25
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:26
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:27
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:28
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:29
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:30
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:31
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:32
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:33
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:34
insert  query update delete getmore command flushes mapped  vsize    res faults
 locked db idx miss %      qr|qw   ar|aw  netIn netOut  conn       time
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:35
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:36
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:37
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:38
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:39
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:40
    *0     *0     *0     *0       0     2|0       0  14.1g  28.3g    40m      0
local:0.0%          0       0|0     0|0   115b      4k      2   19:20:41
```

mongotop

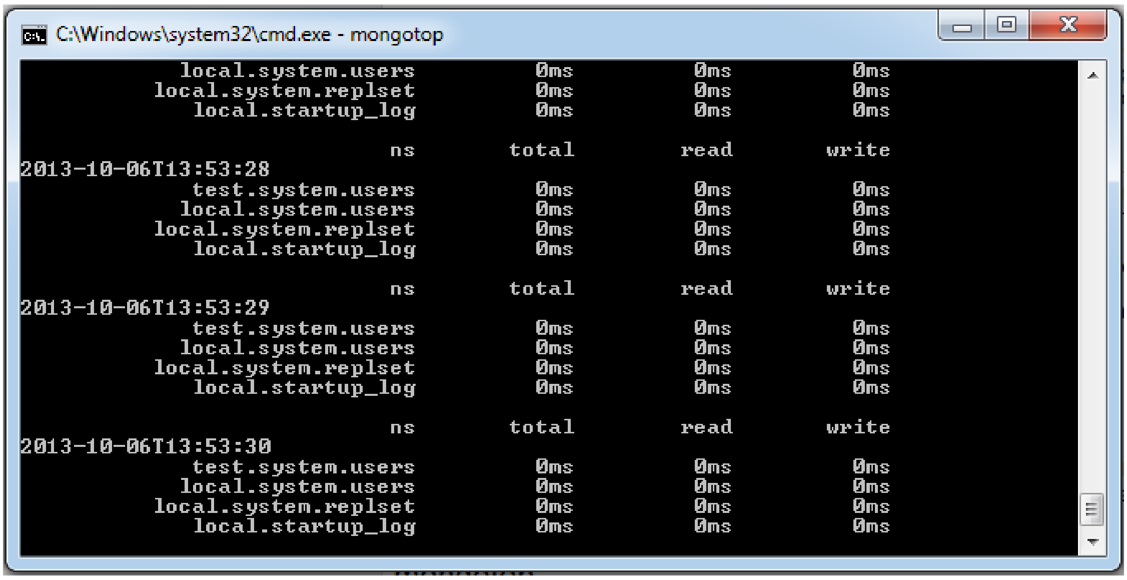
This command track and report the read and write activity of MongoDB instance on a collection basis. By default **mongotop** returns information in each second, by you can change it accordingly.

You should check that this read and write activity matches your application intention, and you're not firing too many writes to the database at a time, reading too frequently from disk, or are exceeding your working set size.

To run the command start your mongod instance. In another command prompt go to **bin** directory of your mongodb installation and type **mongotop**.

```
D:\set up\mongodb\bin>mongotop
```

Output of the command is shown below:



```
           local.system.users          0ms          0ms          0ms
         local.system.replset          0ms          0ms          0ms
            local.startup_log          0ms          0ms          0ms

                           ns        total         read        write
2013-10-06T13:53:28
            test.system.users          0ms          0ms          0ms
           local.system.users          0ms          0ms          0ms
         local.system.replset          0ms          0ms          0ms
            local.startup_log          0ms          0ms          0ms

                           ns        total         read        write
2013-10-06T13:53:29
            test.system.users          0ms          0ms          0ms
           local.system.users          0ms          0ms          0ms
         local.system.replset          0ms          0ms          0ms
            local.startup_log          0ms          0ms          0ms

                           ns        total         read        write
2013-10-06T13:53:30
            test.system.users          0ms          0ms          0ms
           local.system.users          0ms          0ms          0ms
         local.system.replset          0ms          0ms          0ms
            local.startup_log          0ms          0ms          0ms
```

To change **mongotop** command to return information less frequently specify a specific number after the mongotop command.

```
D:\set up\mongodb\bin>mongotop 30
```

The above example will return values every 30 seconds.

Apart from the mongodb tools, 10gen provides a free, hosted monitoring service MongoDB Management Service MMS, that provides a dashboard and gives you a view of the metrics from your entire cluster.

Loading [MathJax]/jax/output/HTML-CSS/jax.js